

PROYECTO FIN DE CARRERA

Presentado a

**LA UNIVERSIDAD DE LOS ANDES
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA**

Para obtener el título de

INGENIERO ELECTRÓNICO

por

Nicolás Hernández Siachoque

Diseño e implementación de un sistema IoT para monitorear calidad del aire.

Sustentado el 14 diciembre de 2021 frente al jurado:

Composición del jurado

- *Asesor:* Fredy Enrique Segura Quijano, Profesor Asociado, Universidad de Los Andes
- *Coasesores:* Juan José García, Universidad de los Andes
Sebastián Sierra, Universidad de los Andes
- *Jurados:* Jorge Mario Becerra Pareja, Profesor de cátedra, Universidad de Los Andes

Contenido

1	INTRODUCCIÓN	3
2	OBJETIVOS	3
2.1	Objetivo General	3
2.2	Objetivos Específicos	3
2.3	Alcance y productos finales	3
3	DESCRIPCIÓN DE LA PROBLEMÁTICA Y JUSTIFICACIÓN DEL TRABAJO	4
4	MARCO TEÓRICO, CONCEPTUAL E HISTÓRICO	5
4.1	Marco Teórico	5
5	DEFINICION Y ESPECIFICACION DEL TRABAJO	6
5.1	Definición.....	6
5.2	Especificaciones	6
6	METODOLOGÍA DEL TRABAJO	8
6.1	Plan de trabajo.....	8
6.2	Búsqueda de información.....	9
6.3	Alternativas de desarrollo.....	10
7	TRABAJO REALIZADO	10
8	VALIDACIÓN DEL TRABAJO	18
8.1	Metodología de prueba.....	18
8.2	Validación de los resultados del trabajo.....	19
8.3	Evaluación del plan de trabajo	21
9	DISCUSIÓN	21
10	CONCLUSIONES.....	21
11	AGRADECIMIENTOS	22
12	REFERENCIAS.....	22
13	APÉNDICES	23

1 INTRODUCCIÓN

Actualmente existen muchas soluciones para medir variables físicas que podrían representar la calidad del aire. Los sistemas integrados permiten medir la concentración de gases, pero en muchos casos no se pueden actualizar con nuevas funciones y no se comunican con otros nodos. Las WSN (redes de sensores inalámbricos) permiten la comunicación entre diferentes nodos pero la información solo está disponible para una autoridad central que puede visualizar el estado de la red. Una solución de IoT permite la comunicación entre dispositivos y está orientada a una arquitectura basada en la nube. La solución propuesta para el monitoreo de la Calidad del Aire en Bogotá se desarrolla con cuatro componentes principales. El primero es el desarrollo del nodo sensor que mide CO, CO2 y material particulado PM2.5 para publicar los valores en un servidor central. El segundo componente es la arquitectura Cloud que incluye un broker MQTT basado en mosquitto, el diseño de la base de datos, un servidor node js para administrar la información y finalmente el software BI para visualizar la información de la base de datos. El tercer componente son las simulaciones de la red con el software cupcarbon. Finalmente, se desarrolló una aplicación móvil en flutter para admitir la configuración de los nodos sensores, los nodos móviles y la visualización de datos en tiempo real.

2 OBJETIVOS

2.1 Objetivo General

Diseñar e implementar una red IoT que permita monitorear la calidad del aire

2.2 Objetivos Específicos

- Diseñar e implementar el nodo de medición con al menos dos indicadores de calidad del aire
- Implementar un sistema con el protocolo MQTT (Red de comunicación y broker)
- Simular por software diferentes nodos en la red de tal manera que sea posible verificar su escalabilidad a diferente número de nodos.
- Aprovisionar una gestión eficiente de los datos y una interfaz web de visualización de datos.
- Implementar aplicación de configuración de nodo de medición que permita establecer rutas basadas en el GPS del dispositivo móvil.

2.3 Alcance y productos finales

A continuación se presentan los principales alcances y productos finales del proyecto de grado

- Nodo de medición funcional de al menos dos variables de calidad del aire. Este nodo de medición cumple con las características de los sistemas IoT como calidad de servicio, tolerancia a fallas, facilidad de uso y la posibilidad de actualización.

- Implementación del sistema en nube (base de datos) que permita manejar la información de los nodos. La recepción de los mensajes de los diferentes nodos por medio de un servidor con un bróker MQTT de mosquitto. El manejo de la base de datos se hace a través de un servidor en node y la visualización a través de software de BI.
- Simulación de varios nodos e indicadores del desempeño de la red al escalar.
- Plataforma para visualizar datos de la red de monitoreo. Tableau permite visualizar la información histórica en la base de datos. La aplicación móvil permite obtener información en tiempo real de los sensores.
- Configuración de nodo fijo a nodo móvil. Por medio de una aplicación móvil que basada en la ubicación del Smartphone puede publicar la información correspondiente a latitud y longitud de nodos móviles.

3 DESCRIPCIÓN DE LA PROBLEMÁTICA Y JUSTIFICACIÓN DEL TRABAJO

En las principales ciudades alrededor del mundo se ha producido un deterioro de la calidad del aire. El 14 de abril del 2021 se firmó en la ciudad de Bogotá un pacto histórico para mejorar la calidad del aire y se fijaron metas a 10 años. Es necesario monitorear diferentes variables para determinar la afectación de gases contaminantes en la ciudad. Se disponen de estaciones meteorológicas cuya información se presenta en el IBOCA (Índice bogotano de calidad del aire), sin embargo, la construcción y mantenimiento de cada una de ellas es de alto costo. Por otro lado, actualmente solo se conoce a nivel de localidad la calidad del aire, en cada una de estas se presentan diferentes fuentes contaminantes que no pueden identificarse solo con una sola estación meteorológica. Se requiere entonces construir una red de monitoreo de gases efecto invernadero y material particulado con la característica de bajos costos de inversión y mantenimiento. Hace falta explorar nuevos dispositivos que se acoplen a esta red y que de manera económica contribuyan con información de tal manera que aumente la densidad de puntos de medición en la ciudad.

Con el desarrollo tecnológico en el área de microprocesadores, sensores de bajo costo y mejora en las comunicaciones inalámbricas, se ha potenciado el desarrollo de las soluciones IoT. Los servicios en la nube han permitido que soluciones escalables sean cada vez más fácil de implementar sin la necesidad de realizar altas inversiones de capital en centros de datos. Debido a esto, los sistemas IoT son la solución que actualmente se desarrolla en aquellos ámbitos donde se necesita monitorear variables en ciudades inteligentes. Es importante el desarrollo de estas soluciones ya que no solo aplican para el monitoreo de calidad del aire, también permiten el desarrollo de una ciudad más conectada donde cada ciudadano es partícipe de la información que se genera. Políticamente es relevante ya que existe un interés de la administración distrital en obtener información que permite implementar políticas más assertivas para disminuir los índices de contaminación.

4 MARCO TEÓRICO, CONCEPTUAL E HISTÓRICO

4.1 Marco Teórico

Calidad del aire: la calidad del aire está dada por diferentes mediciones de gases y material particulado. En la literatura se encuentran principalmente el monóxido de carbono CO, dióxido de carbono CO₂ y el material particulado PM2.5 y PM10. En la ciudad de Bogotá, la RMCAB (Red de Monitoreo de Calidad del Aire de Bogotá) obtiene información de la concentración de material particulado (PM10, PST, PM2.5), gases contaminantes (SO₂, NO₂, CO, O₃) y de las variables meteorológicas de precipitación, velocidad y dirección del viento, temperatura, radiación solar, humedad relativa y presión barométrica por medio de 13 estaciones fijas y 1 estación móvil.

Material particulado: indica las partículas sólidas presentes en el aire, PST corresponde al total de partículas suspendidas, PM2.5 indica aquellas con diámetro menor a 2.5 micrómetros y PM10 inferior a 10 micrómetros. Estas partículas generan una afectación al ser humano ya que son capaces de llegar a los pulmones y generar enfermedades.

Protocolo MQTT (message queue telemetry transport protocol):

es un protocolo para la comunicación de dispositivos que funciona bajo la arquitectura publicar/subscribir. Esta arquitectura permite que el protocolo sea más liviano en el uso de recursos tanto de GPU como de red a comparación de una arquitectura cliente-servidor. El cliente publicador contiene sensores que obtienen las medidas que se publican al broker para ser procesadas desde este. El cliente subscriptor hace uso de la información disponible en el broker.

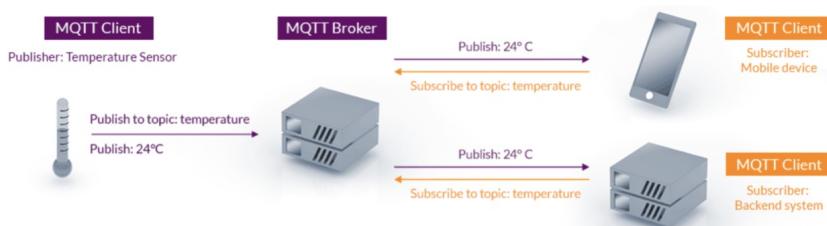


Ilustración 1. Arquitectura protocolo MQTT [12]

Antecedentes externos

En [3], se presenta la implementación de una red de monitoreo de calidad de aire en un campus universitario. En este artículo se muestra un buen desempeño del uso del protocolo MQTT en términos de tiempo y ahorro económico de los dispositivos utilizados. Como complemento a futuro, los autores establecen la necesidad de integrar una mejor interfaz para poder visualizar y manipular los datos con el fin de realizar análisis más detallados.

En [4] se realiza un monitoreo que incluye el servicio de nube de AWS. En este caso, al implementar un sistema de nube, es posible analizar a mayor profundidad los datos y hacer uso de herramientas

de AWS que eliminan ruido del sensor y permiten el uso de interfaces gráficas como Kibana, además se generan alertas vía SMS al superar ciertos niveles en los sensores.

En [5], a diferencia de incorporar solo los sensores de calidad de aire, se incluye un modulo de GPS con el fin de tener nodos móviles. En este caso se utiliza una Raspberry Pi 3 y como proveedor de nube a Google Firebase. En este caso la utilidad de nodos móviles se encuentra en que pueden generar un mapa de rutas de transito de vehículos públicos o privados. Es posible rastrear varios puntos de una ciudad empleando el mismo nodo. Estos datos pueden ser de utilidad para identificar zonas con mala calidad del aire y poder intervenir en ellas.

Antecedentes locales

En [6] se implementó un sistema de comunicación M2M (Machine to machine) con el fin de representar una red de IoT con el protocolo MQTT en la Universidad de los Andes. El proyecto de grado emplea como nodos a un Arduino, un broker en máquina virtual y un nodo en un teléfono Android para visualizar y activar o desactivar el sensor.

5 DEFINICION Y ESPECIFICACION DEL TRABAJO

5.1 Definición

El trabajo realizado busca la conformación de una red de monitoreo de calidad del aire que permita mejorar en gran medida la resolución espacial y temporal de la información que actualmente se puede recolectar. La red busca integrar no solo a estaciones meteorológicas sino a cualquier agente desde el nivel de cada ciudadano, pasando por organizaciones con instalaciones en la ciudad e inclusive llegar a escala de barrios o localidades. Este trabajo aplica para todos los ciudadanos interesados en conocer el estado de calidad del aire, como ciclistas, asociaciones de conjuntos residenciales, universidades, barrios, localidades y el mismo distrito por medio de la secretaría de ambiente. Como consideración económica se encuentra que el desarrollo de prototipos físicos está limitado a bajos costos ya para aumentar la resolución espacial, es necesario contar con cientos o miles de sensores. Además de esto, el proyecto puede utilizar los recursos en el laboratorio de ingeniería electrónica de la universidad, sin embargo la adquisición de materiales para los prototipos es asumida por el estudiante. Finalmente, se considera que el proyecto pueda tomarse como referencia a futuro y así concretar una mejor red de monitoreo, basado el trabajo de toda una comunidad académica.

5.2 Especificaciones

La solución debe contener las principales características de un sistema IoT.

Calidad de la información: Se espera que la red maneje datos de calidad que permitan generar una representación apropiada de las condiciones de calidad del aire en la ciudad.

Escalabilidad: Para dar soporte a una gran cantidad de nodos, la solución debe ser capaz de crecer para permitir el ingreso de nuevos usuarios. De manera aceptable se espera un soporte al orden de cientos de nodos, que permiten distribuirse espacialmente de tal manera que se pueda cubrir un barrio de 20 por 20 cuadras con un nodo por cada cuadra aproximadamente. De manera ideal se espera escalar al orden de miles de nodos ya que para cubrir un área de 200 por 200 cuadras se requerirían unos 40.000 nodos.

Tolerancia a fallas: Se espera que la red sea capaz de responder a diferentes problemáticas que se pueden presentar. La principal falla que puede presentarse es perdida de conectividad entre los nodos, para ello el sistema debe continuar censando y almacenar la información de tal manera que una vez se solucione el problema de conectividad, las medidas puedan almacenarse en la base de datos. Para definir un nivel aceptable se espera que los dispositivos puedan operar sin conexión por un día que se considera tiempo suficiente para solucionar cualquier inconveniente presentado. Sin embargo idealmente se esperaría poder llegar a una semana de operación sin conexión.

Integración de diferentes tipos de dispositivos: Las soluciones IoT deben dar soporte a incluir dispositivos con mejores según los avances tecnológicos. En el caso de la problemática de calidad del aire, es de gran importancia el nivel de contaminación de agentes móviles como lo son buses de transporte público o las rutas que recorren automóviles o vehículos de carga. Estas fuentes contaminantes móviles requieren de nodos que publiquen información de latitud longitud. En vista que actualmente hay sensores GPS en la mayoría de celulares inteligentes es posible integrar a los smartphones como dispositivos que contribuyen a la red que no necesariamente comparten la misma tecnología que los nodos sensores. Para cumplir con este requerimiento se espera un soporte de nodos móviles en la red.

Configuración automática: El objetivo es reducir los costos de mantenimiento que pueden tener los nodos sensores, por tal motivo se espera que usuarios sin conocimientos técnicos sean capaces de configurar y manipular los nodos sensores.

Procesamiento de datos: Se espera poder hacer uso de los datos recolectados para visualización de ellos, estadísticas e inclusive permitir un soporte a futuro de modelos predictivos. Para satisfacer el requerimiento se espera una interfaz de visualización y que la solución cuente con alternativas para a futuro poder implementar diferentes modelos analíticos o predictivos.

Manejo energético: Se espera un manejo energético eficiente para garantizar una gran autonomía de los dispositivos de censado. Este manejo energético espera cumplirse a través de los protocolos empleados y las optimizaciones en software de los sensores. Se espera que la solución cuente con tácticas de ahorro de consumo que se puedan implementar a futuro por medio de una actualización.

Seguridad: La seguridad en los sistemas IoT es de gran importancia ya que se maneja un alto volumen de nodos y comunicaciones. Sin embargo, este requerimiento puede generar conflicto con otros de los requerimientos anteriores y en vista que el objetivo es una red pública de monitoreo, no se toma como uno de los requerimientos principales. Sin embargo, sí se espera que la solución esté en capacidad de integrar tácticas de seguridad a futuro.

6 METODOLOGÍA DEL TRABAJO

Al inicio del semestre se realiza un cronograma de actividades agrupadas por diferentes hitos. Cada semana tiene unas tareas asignadas que son presentadas en el seminario de “Técnicas y Tecnologías de diseño electrónico” del CMUA. En las reuniones se realiza una retroalimentación del trabajo realizado y se da el espacio para solucionar o aclarar inquietudes respecto al trabajo de la siguiente semana. Una vez se completan las primeras 8 semanas, se realiza una recopilación del avance de estas para identificar retrasos en el cronograma y replantear la organización de actividades de las siguientes 8 semanas (en caso de ser necesario).

6.1 Plan de trabajo

Las actividades realizadas se agrupan en 5 categorías principales.

- A. Diseño y desarrollo del nodo sensor: En este componente se realizan las tareas de diseño del nodo de censado. Incluye la selección y calibración de sensores de gas y polvo, esquemático y diseño de PCB. Programación de la tarjeta para transmitir la información censada, permitir la configuración del dispositivo, técnicas de recuperación ante fallas, entre otras actividades. El desarrollo de las actividades de diseño e implementación hardware se realizaron en la primera mitad del semestre, las funcionalidades de software empezaron su desarrollo entre semanas 7 y 8 con finalización entre semanas 12 y 13. La duración total de esta categoría fue de 13 semanas.
- B. Diseño y desarrollo de la arquitectura de la red en la nube: En ella se establecen las actividades del diseño de la red basada en MQTT, la implementación de un bróker basado en mosquitto, el diseño de la base de datos para almacenar información de la red, un servidor node para acceder a la base de datos y la herramienta de visualización de BI Tableau. Estas actividades van de manera transversal a la implementación de todos los servicios en la nube de GCP. Esta categoría inició su desarrollo a partir de la quinta semana del semestre y culminó en la semana 14 con una duración total de 9 semanas.
- C. Simulaciones: Las simulaciones realizadas en CupCarbon permiten el apoyo para pruebas en todo el proceso de desarrollo de la red. Por tal motivo se realizaron diferentes simulaciones, específicamente una a mitad de semestre en la semana 8 y otra al finalizar el semestre a partir de la semana 14. En total se cuenta con una duración de 3 semanas.
- D. Aplicación móvil: Para el desarrollo de la aplicación fue necesario conectar el dispositivo móvil por los diferentes protocolos(BLE, HTTP, MQTT) tanto de red como de aplicación configurar la interfaz para interactuar. Como interacciones principales está la configuración de nodo sensor nuevo, administrar nodo sensor actual (incluye nodo móvil) y visualización de históricos y datos en tiempo real. Esta etapa inicia su desarrollo en la sexta semana del semestre y hasta finalizar el proyecto de grado. La duración total fue de 10 semanas.
- E. Documentación. En esta categoría se incluye actividades de revisión del estado del arte que se enfocan en las tres primeras semanas del semestre, sin embargo son transversales a las demás actividades. Además se cuenta con la documentación de los repositorios con los archivos de cada herramienta que se utilizó. Finalmente es importante tener en cuenta las actividades de seguimiento como lo son las presentaciones semanales en el seminario de “Técnicas y

Tecnologías de diseño electrónico”, los informes de propuesta de proyecto de grado, seguimiento de mitad de semestre y el presente documento final.

A continuación se presenta el cronograma con las respectivas convenciones de las categorías.

Actividad	Semanas																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Revisión estado del arte																	
Adquisición y prueba de sensores																	
Diseño sensor																	
Prototipado del sensor																	
Implementación PCB 2 o 3 sensores																	
Implementación software del sensor																	
Aplicación de configuración de dispositivos																	
Prueba y calibración sensores																	
Diseño del broker																	
Implementación broker																	
Pruebas de la red. Simulación de sensores																	
Configurar Base de datos																	
Crear servicios a utilizar de la nube																	
Establecer el API de consulta																	
Implementación interfaz de visualización																	
Interfaz de visualización de nodos móviles																	
Documentación, informe y presentación PG																	

Ilustración 2. Cronograma

6.2 Búsqueda de información

Las fuentes con información nueva que se empleó para el presente proyecto de grado fueron las siguientes.

- ESP-IDF Programming Guide: Posee la documentación del api y especificaciones de hardware de la tarjeta de desarrollo ESP-32 que se empleó para el nodo sensor.
- Eclipse Mosquitto: Es el bróker open source que se empleó para la red en el protocolo MQTT. Hace parte de Eclipse Foundation en la que se presentan diferentes proyectos colaborativos open source.
- Datasheets para los diferentes sensores. La selección previa de candidatos a sensores de variables de calidad del aire se basó en publicaciones de trabajos anteriores de IoT en IEEE o indexados en SCOPUS, después de una preselección de diferentes sensores, se seleccionaron aquellos disponibles para adquirir en Colombia. Respecto a la disponibilidad principalmente se basó en el proveedor Sigma Electrónica, sin embargo se tuvo en cuenta otros proveedores. Por otro lado, la consulta en los datasheets de los diferentes sensores ayudó a determinar aquellos con los que se trabaja de manera posterior.
- Documentación Flutter SDK y pub.dev: Para el desarrollo de la aplicación móvil. El sdk de flutter y dart se empleó junto al uso de algunas librerías disponibles en el repositorio de paquetes pub.dev.
- Google cloud documentation: Permite la implementación de la arquitectura en la nube.

Respecto a la preparación académica previa al proyecto de grado se encuentra un manejo previo de herramientas como Eagle, programación en lenguajes como C, javascript y dart. Conocimientos previos en el diseño de bases de datos relacionales con postgres. Conceptos básicos de los principios de operación de sensores como cambios de resistencia en materiales al exponerse a reacciones químicas o efectos fotoeléctricos.

6.3 Alternativas de desarrollo

La principal alternativa es el uso de un protocolo de comunicación en capa física diferente al Wi-Fi o al BLE. Es posible implementar el nodo sensor con otra tecnología, sin embargo incrementaría la complejidad de este ya que sería necesario adquirir nuevas antenas que den soporte a otros protocolos. Otros protocolos pueden tener un desempeño mejor o peor según los requerimientos de la red, pero el desarrollo de ellos trae un tiempo de experimentación con la nueva tecnología que además de ser extenso, tiene un alto riesgo de extenderse en el cronograma. Por tal motivo, fue necesario acotar a los protocolos disponibles en la placa ESP-32 y con ello completar las actividades de todas las categorías especificadas en el plan de trabajo. Wi-Fi es un protocolo con cobertura en la mayor área de la ciudad, es configurable de manera sencilla y existe un trabajo previo en la literatura y diferentes proyectos que inclusive poseen librerías para un desarrollo más rápido. El Bluetooth de baja energía (BLE) tiene un consumo energético hasta 10 veces menor que la comunicación tradicional por Bluetooth, esto es importante para el requerimiento de consumo energético.

7 TRABAJO REALIZADO

A. Nodo sensor:

Para el nodo de censado se trabajo tanto en el diseño e implementación del hardware como en la programación del software.

Hardware

En primer lugar se realiza la selección de variables a medir. Para ello se toma en cuenta los sensores empleados en la literatura y se acotan por su disponibilidad en Colombia y los costos máximos a un valor menor a los 50.000 pesos colombianos, ya que la adquisición de ellos es asumida por el estudiante. Con base en las limitaciones y con el objetivo de obtener la mayor calidad de servicio en las mediciones. Se definen los sensores MQ-7 y MQ-135 para sensar diferentes concentraciones de CO y CO₂ respectivamente y el GP2Y1010AU0F para medir concentración de material particulado (polvo).

Sensores electroquímicos MQ:

Ambos sensores (MQ-7 y MQ-135) funcionan con el mismo principio de operación. Miden el cambio de la resistencia que posee un sustrato al reaccionar con diferentes gases. Estas reacciones de oxidación/reducción son aceleradas por medio de un calentador interno en el sensor. El sustrato

utilizado para medir el cambio de resistencia es el dióxido de estaño SnO₂. Los datasheets con la información en que se basó para la calibración se encuentran en [9] y [10]

Se ajusta la curva característica de tal manera que $R_s/R_0 = a \cdot (ppm)^b$

Basado en un trabajo previo de Félix Marco Millán [1] se toman los valores de $a = 5.5973021420$ y $b = -0.365425824$.

$$\frac{R_s}{R_0} = a \cdot (ppm)^b \quad R_0 = \frac{R_s}{a \cdot ppm^b}$$

R_0 se toma después de calibrar el sensor 3 veces en espacio cerrado sin acceso a contaminación en el exterior. El valor de esta resistencia obtenido fue de 7.2 kΩ

Por otro lado se tiene la resistencia R_s que está definida por

$$R_s = \frac{1024R_L}{analogRead} - R_L$$

Con un R_L de 20 kΩ

Finalmente, para obtener las partes por millón se emplea la ecuación

$$ppm = \sqrt[b]{\frac{1024R_L/analogRead - R_L}{R_0}} \quad a$$

Sensor de polvo:

El dispositivo utilizado es el GP2Y1010AU0F que es un sensor óptico que funciona con un fototransistor. Es importante tener en cuenta que este sensor debe emitir luz por medio de un diodo infrarrojo. Esta emisión de luz se hace de manera periódica y se establece un simple time de 280 ms para hacer la medición de partículas de polvo. Este sensor es capaz de detectar partículas de hasta 0.8 micras por lo que se puede emplear para definir el nivel de material particulado 2.5 (de hasta 2.5 micras). El esquema interno del sensor amplifica la salida del fototransistor. La sensibilidad es de 0.5v por 0.1mg/m³. La salida de este sensor permite definir la densidad de material particulado según la curva característica donde los valores son proporcionales a la lectura analógica por parte de la placa ESP-32. Los parámetros de la curva característica se toman directamente del datasheet [11].

Selección de la tarjeta de desarrollo

Se selecciona la tarjeta de desarrollo ESP-32 ya que esta posee herramientas útiles para soluciones IoT. Entre las utilizadas se encuentran los protocolos WiFi y BLE, manejo de una memoria no volátil para carga de preferencias, uso de un sistema de archivos simple y acceso a un sistema operativo de tiempo real.

El desarrollo de la PCB se realizó en Eagle donde primero se obtuvo las librerías correspondientes a los componentes que se emplean para generar un esquemático y posteriormente los archivos del circuito impreso.

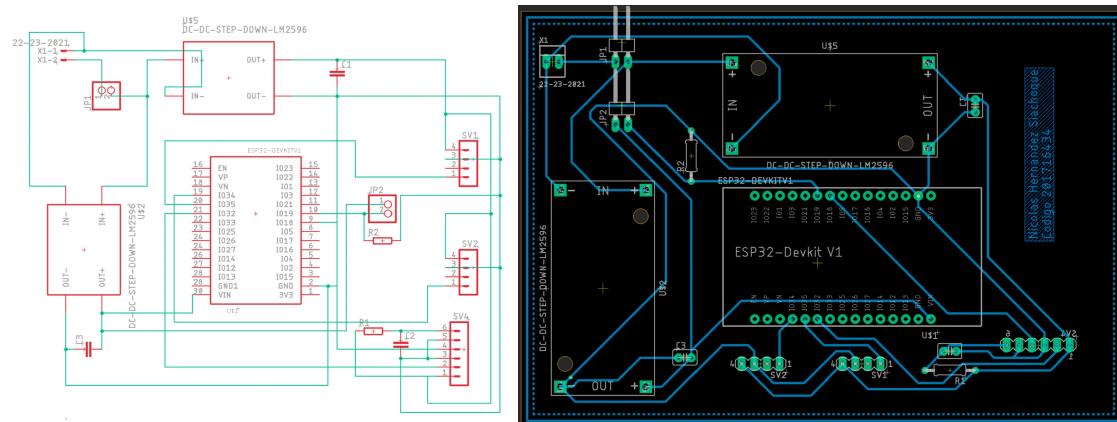


Ilustración 3. Esquemático y PCB del nodo sensor

El costo total del hardware de cada nodo sensor fue aproximado de \$100.000 pesos colombianos (\$25 dólares). Los materiales empleados tienen costos menores si es a una producción a escala, por otro lado no se asumió un costo para la fabricación de la PCB ya que este fue proporcionada por el laboratorio de circuitos impresos. Es importante aclarar que las piezas utilizadas son importadas, por tal motivo fluctúan su valor según el precio del dólar. Existen componentes que al importarse directamente de China son más económicos pero existe un tiempo de llegada al país de al menos 3 meses, por ello se decidió adquirirlos con proveedores nacionales a pesar de tener un costo mayor.

Software:

El software se programó en C utilizando el api del IDF oficial para la tarjeta esp-32 junto con librerías para protocolos MQTT y HTTP disponibles del IDE de Arudino. Entre las principales funciones de software que se desarrollaron se encuentra.

Censo y publicación de variables: Se realizan los cálculos para convertir la señal análoga a las unidades de cada variable de medición. En el caso de las variables de concentración de gases se hace la conversión a partes por millón, mientras que para el material particulado se convierte a mg/m³. Los datos se envían por medio del protocolo MQTT al bróker principal. Cada variable posee su tópico correspondiente en el que se publica el dato obtenido.

Conectividad y manejo de errores: Los diferentes dispositivos en una red de internet de las cosas deben responder de manera adecuada a posibles fallos que se pueden presentar. Uno de ellos es la perdida de conexión por la cual se envían los datos del sensor. En el caso del nodo desarrollado se emplea conexión por Wi-Fi y en casos donde no se encuentre cobertura de la señal se definió que las variables sensadas se almacenarán para enviarlas a la base de datos una vez se recupere la conexión. Para la persistencia de la información se utiliza un sistema simple de archivos ya existente en la ESP-32 que se denomina como SPIFFS. Se genera un archivo según la fecha y variable a registrar, cada medición se guarda como una dupla de hora y valor. Esto permite que la información se almacene en la memoria flash integrada y al recuperar conexión envíe al servidor los datos por medio del protocolo HTTP.

Persistencia de configuraciones de cada dispositivo. Existen diferentes variables que deben almacenarse para que al reiniciar el dispositivo puedan cargarse nuevamente. Entre ellas se encuentra el identificador del dispositivo, dirección IP y puerto del bróker MQTT, dirección IP del servidor node, nombre SSID de la red wifi con su respectiva contraseña, tipo de nodo (móvil o estático) y versión del software. Para ello se emplea una memoria no volátil NVS que se encuentra integrada en la memoria flash de la ESP-32, las variables almacenadas en esta memoria son de gran cantidad de lecturas pero bajo número de escrituras.

Conectividad por medio de Bluetooth de baja energía BLE para configurar dispositivo: Se define el servicio Device Information para incluir todas las variables que serán manipuladas por la aplicación móvil, cada variable representada en una Bluetooth characteristic del servicio ya mencionado. Estas características tienen definidos unos permisos de escritura y lectura según la sensibilidad de los datos en cada una de ellas. El software invoca funciones callback si hay cambios en alguna de las características para poder aplicarlos si es necesario sobre la NVS o ejecutar funciones específicas como reiniciar el dispositivo.

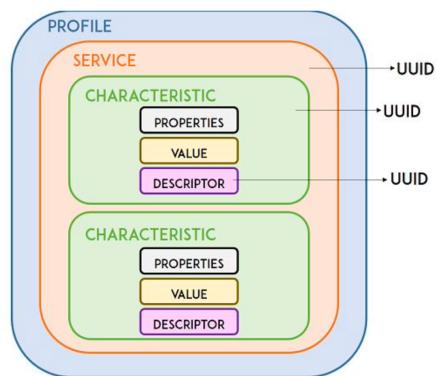


Ilustración 4. Estructura de servicios y características en BLE.[13]

Uso de un sistema operativo en tiempo real FREERTOS: La placa de desarrollo ESP-32 viene con el sistema operativo FREERTOS incluido, que tiene la característica de ser open source. Internamente se encarga de ejecutar las actividades de conectividad y otras esenciales para el correcto uso de los recursos, en el procesador número uno y el programa principal en el procesador dos. Existen diferentes funciones que se deben realizar de manera concurrente y se pueden ejecutar en el mismo procesador, que en este caso es el número dos.

Para ello se definen estados de las actividades o tareas que comparten diferentes recursos. Los estados permiten definir que tarea está en ejecución y cada tarea puede tener una prioridad de ejecución diferente. La comunicación entre tareas se puede realizar a través de colas y es importante tener presente el acceso a recursos compartidos (WiFi, SPPIF, GPIO, etc). Las actividades que se definieron para el sistema operativo son la verificación del estado de conexión a internet, escritura en sistema de archivos en caso de perdida de conexión y el censado de las diferentes variables.

B. Arquitectura de la red en Cloud

Para manejar la información de los sensores se emplea el protocolo en capa de aplicación MQTT. El bróker (implementación open source de mosquitto) que se encarga de administrar los mensajes y los tópicos es una máquina virtual de Google Cloud Platform. A este bróker se comunican todos los agentes que requieren información como los nodos sensores, dispositivos móviles y el servidor Node JS. Este servidor node JS se encarga de la comunicación a la base de datos, responde a solicitudes de la aplicación móvil y persiste los datos guardados por los nodos sensores cuando recuperan conexión y que no pudieron transmitirse por MQTT en tiempo real. A continuación se presentan detalles de algunos componentes que se emplearon.

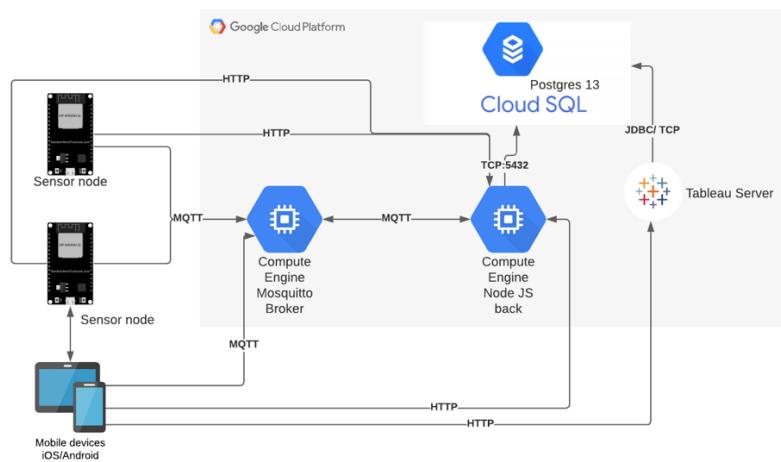


Ilustración 5. Arquitectura de la red IoT en la nube

Base de datos:

Se seleccionó una base de datos relacional SQL para el almacenamiento de la información. Se consideró el uso de bases de datos no SQL diseñadas para series de tiempo pero no se utilizaron debido a que al no tener un esquema definido es propenso a tener inconsistencias al almacenar información de rutas, organizaciones, entre otros. El modelo relacional de la base de datos fue el siguiente

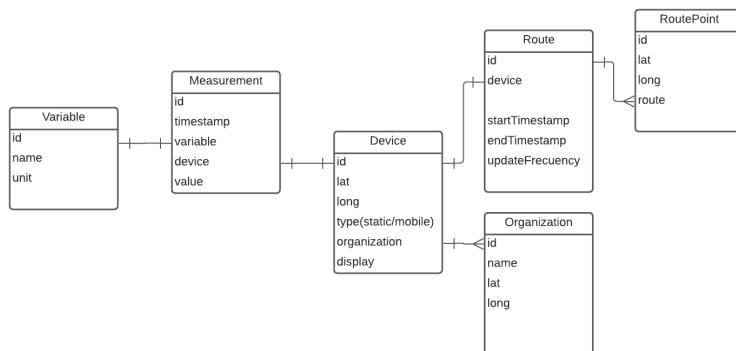


Ilustración 6. Esquema de la base de datos

Este diseño permite que a futuro se agreguen otras variables que afectan a la calidad del aire. También agrupa los dispositivos en organizaciones de tal manera que se pueda manejar diferentes niveles de granularidad, por ejemplo los dispositivos de una institución como una universidad, a nivel de un barrio o localidad.

Las especificaciones para las máquinas virtuales del bróker y del cliente node son las siguientes. E-2 micro con 2 CPUs virtuales Intel Broadwell 1GB de memoria zona us-east1-b sistema operativo Ubuntu 20.04 que tienen un costo inferior de \$7.5 USD mensuales según la demanda de uso cada una.

Broker mosquitto

Este bróker sigue la implementación open source de mosquitto. Sin embargo tiene modificadas unas preferencias para acoplarlo a la red que se diseño. Para un desarrollo rápido del proyecto de grado fue necesario determinar niveles de importancia para cada atributo de calidad, en el caso de la seguridad se definió una menor importancia respecto a otros aspectos por lo que se decidió no implementar el cifrado en el protocolo MQTT ni la restricción de usuarios. Al no incluir el cifrado se permite un desarrollo más ágil de la solución ya que se pueden emplear herramientas como wireshark para identificar un correcto funcionamiento de la transmisión de mensajes. Además evita la necesidad de emitir certificados ssl que de todas maneras no contaría con el aval de alguna autoridad de certificación sin incurrir en costos económicos adicionales. El bróker entonces elimina la necesidad de incluir certificados y usuarios que accedan a la red MQTT, por este motivo también se descartó la aplicación de bróker nativa de GCP.

Servidor Backend Node

Se emplea el entorno de ejecución Node JS que permite una programación en JavaScript y el uso de librerías del administrador de paquetes npm. Este servidor se subscribe al servidor con el bróker para poder persistir los datos que llegan de los sensores. Para comunicarse con la base de datos utiliza la librería PG y toma las credenciales de la base de datos de variables de entorno. Por otro lado, ofrece el API REST de consulta para la aplicación móvil, mediante esta se exponen servicios GET, POST, PUT.

Tableau

Se selecciona como plataforma para visualizar la información en la base de datos. Este software permite realizar consultas y crear campos calculados para presentar hojas de visualización o dashboards. Principalmente se emplea para generar un mapa de calor de la información en la red de los diferentes históricos.

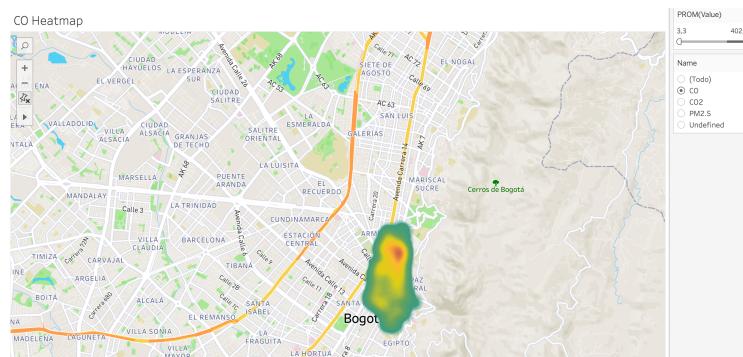


Ilustración 7. Mapa de calor en Tableau

C. Aplicación móvil

La aplicación móvil se desarrolló en el framework de desarrollo de aplicaciones Flutter. La principal ventaja de este framework es que permite generar aplicaciones para diferentes plataformas de manera sencilla. Se emplea el API de google maps de GCP para tanto iOS como para Android. Esto con el fin de brindar mapas a la interfaz de la aplicación. Se emplean paquetes de dart con el fin de utilizar los servicios de localización, uso de protocolos como http y mqtt, generar visualizaciones de gráficas, entre otras funciones. El detalle de los paquetes utilizados se encuentra en el archivo *pubspec.yaml* del repositorio de la aplicación.

La aplicación cuenta con diferentes módulos. El primero muestra opciones de conexión a nodos sensores, bien sea por medio de Bluetooth o directamente consultándolo por su organización y su identificación. Al conectarse por medio del identificador, solo es posible consultar la información en tiempo real de las variables y el histórico. La información en tiempo real funciona por medio del protocolo MQTT donde el celular se suscribe a los tópicos donde el sensor publica la información. Por otro lado, la información de históricos se obtiene por consultas al API REST del servidor node. Si por el contrario, se conecta por bluetooth, es posible reconfigurar el dispositivo (Por ejemplo, cambiar la red WiFi) realizar acciones sobre el dispositivo como reinicio o restaurarlo. Otra funcionalidad exclusiva para conexión por bluetooth es conocer el estado del dispositivo, en este caso se puede consultar la señal de Wi-Fi, estado de conectividad y almacenamiento libre del dispositivo. La comunicación con el dispositivo es por medio de BLE y se emplean las características disponibles en el servicio Device information. Finalmente es posible configurar el nodo móvil, permite crear nuevas rutas y visualizar antiguas (Tanto el mapa como las graficas de cada variable censada). Para el funcionamiento del nodo móvil, la aplicación publica por el protocolo MQTT la información de latitud y longitud del GPS integrado al celular.



Ilustración 8. Capturas de la aplicación

Existe un último módulo para que el usuario consulte el estado actual de la red de monitoreo. Para ello ofrece dos herramientas. La primera es una interpolación de los datos de la red específicamente para la ubicación del celular. Para ello toma todos los nodos en un radio inferior a 500 metros y obtiene los datos promedio de la última hora para calcular estimaciones en cada variable. Por otro lado, el usuario puede consultar el mapa de calidad del aire con el fin de visualizar el estado de contaminación en toda

la ciudad para un intervalo de horas y una fecha definida. Puede filtrar por tipo de variable y visualizar para un nodo específico el histórico de mediciones o consultar los datos en tiempo real.

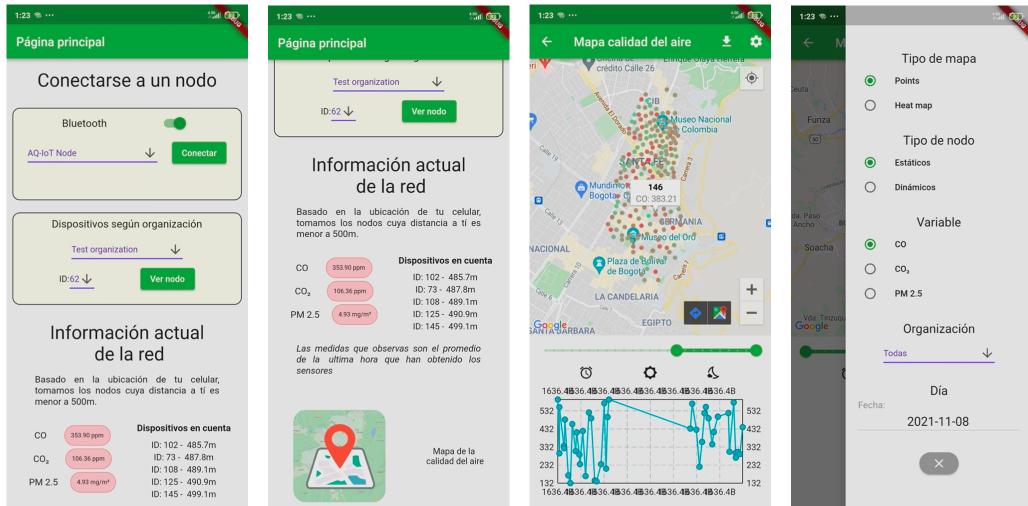


Ilustración 9. Visualización del estado actual de la red

D. Simulaciones

Se realizaron dos simulaciones principales en el software de cupcarbon. Una primera simulación con 4 nodos (móviles y estaticos), en la que solo se verifica funcionamiento correcto de la red. La segunda simulación tiene en cuenta desde 50 hasta 200 nodos que publican datos aleatorios. Esta simulación tenía como objetivo llegar a los mil nodos, sin embargo, por restricciones de la aplicación que generaba las conexiones, no fue posible llegar a este objetivo.

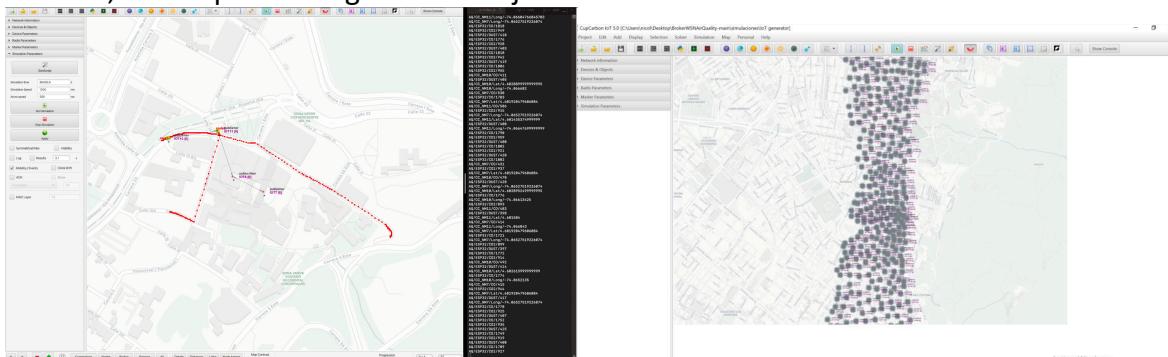


Ilustración 10. Simulaciones en CupCarbon

E. Documentación

Todo el contenido se encuentra documentado en presentaciones semanales y los documentos de propuesta, mitad de semestre y documento final. Además la información se encuentra consolidada en una organización de github. En ella se encuentran 3 repositorios con códigos fuente, archivos, entre otros documentos. El primer repositorio es esp32 donde se encuentra el código en C de la tarjeta esp-32 y los archivos de esquemático y PCB realizados en Eagle. El repositorio aq_iot_flutter posee el código

fuente de la aplicación móvil. Finalmente el repositorio BrokerWSNAQ posee el código fuente del back en node, sentencias SQL para la base de datos y los archivos de simulación de cup carbón.

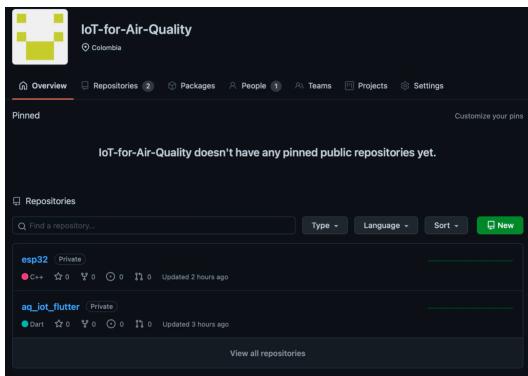


Ilustración 11. Organización con repositorios en GitHub

8 VALIDACIÓN DEL TRABAJO

8.1 Metodología de prueba

Prueba de funcionamiento de nodos estáticos:

Para estos dispositivos se definió un tiempo de muestreo de 5 segundos. Se utilizan dos nodos físicos de manera simultanea y desde la aplicación móvil se verifica la conexión en tiempo real. Se emplea un dispositivo Android físico (Xiaomi Mi 8 Android 10) que hace funciones de configuración de los nodos ya que este tiene acceso al sensor bluetooth. Por otro lado se emplea un dispositivo iOS (iPhone 13 iOS 15) simulado por Xcode que permite consultar los datos en tiempo real de los nodos. Para las pruebas de conectividad se procede a establecer la conexión de un nodo sensor a la red de internet compartida por el teléfono Android. Posteriormente se desconecta la función para compartir internet y en la aplicación se busca el indicador de perdida de conexión (Adicionalmente un led en la placa indica la perdida de conexión). Finalmente se reestablece la conexión a la red de internet y se verifica tanto en base de datos como en la aplicación móvil que los datos durante el periodo de desconexión se persistan correctamente.

Prueba de funcionamiento de nodos móviles:

Se establece un nodo móvil asociado a el dispositivo Android. Por medio de las opciones de desarrollador se permite el uso de aplicaciones simuladoras de ubicación que permiten emular ubicaciones falsas en el dispositivo. Se seleccionó la aplicación Mock GPS cambiar la ubicación antes de iniciar una nueva ruta. Una vez se inicia la ruta, la aplicación Mock GPS permite emular los movimientos de un automóvil. La frecuencia de muestreo de latitud y longitud es igual que la de censado de las variables, cada 5 segundos. Finalmente se verifica tanto en base de datos como en la aplicación la información de la ruta creada.

Pruebas de simulaciones y escalabilidad de la red:

Por medio del software cupcarbon se establecen 200 nodos distribuidos en la ciudad de Bogotá. La aplicación se encarga de enviar los datos aleatorios por cada nodo al bróker desplegado en la nube. Finalmente se verifica el comportamiento de la red VCP en la nube y el uso de la CPU de las máquinas virtuales.

8.2 Validación de los resultados del trabajo

Para las pruebas del funcionamiento de los nodos móviles se verificó el funcionamiento de los sensores, el registro en la base de datos tanto de nuevos dispositivos como de la toma de variables. En la aplicación móvil se verifica el funcionamiento en tiempo real de las variables de medición.



Ilustración 12. Resultados de nodos estáticos

Para los nodos móviles se validó desde la misma aplicación la visualización de una ruta con sus respectivas gráficas de contaminación.

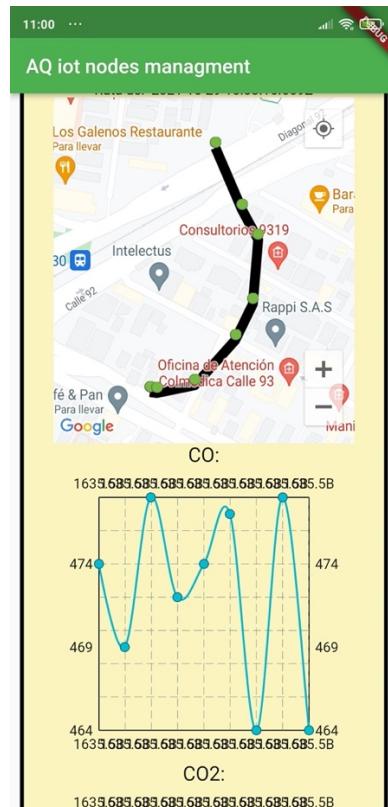


Ilustración 13. Resultados de nodos móviles

Respecto a la escalabilidad de la red se monitoreo el uso de la VCP de GCP y el uso de CPU de las máquinas virtuales. Debido a las restricciones de simulación solo se simularon 200 nodos para los cuales el uso de recursos no superó el 20% con la instancia de máquina virtual más económica. Para el soporte de miles de nodos es posible llegar a saturar estas máquinas que instancian nuevas máquinas virtuales que apoyen al procesamiento de la información.

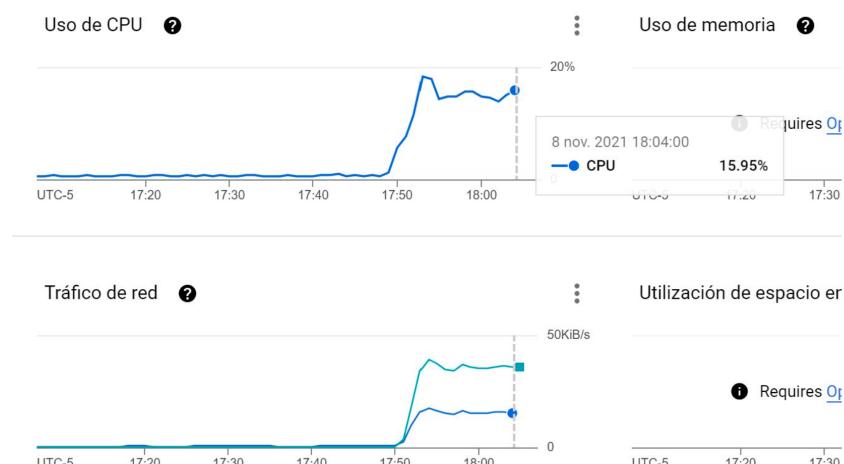


Ilustración 14. Uso de recursos en simulación de 200 nodos.

8.3 Evaluación del plan de trabajo

Se completaron las actividades esperadas y se cumple con los atributos mínimos de calidad. Sin embargo, la cantidad de actividades planteadas requieren un tiempo muy ajustado para llevarlas a cabo en solo 16 semanas del semestre. A manera de reflexión, es importante tener en cuenta tiempos más largos para incluir espacio a imprevistos como errores en el desarrollo de cada actividad. Al no ser tenidos en cuenta estos espacios, fue necesario un trabajo extra cada semana y movimientos en el cronograma durante el semestre. Por otro lado, definir una gran cantidad de actividades afecta en la calidad de las mismas. Es posible haber profundizado más en cada una de ellas, sin embargo, al buscar constituir una red IoT desde cero, el trabajo realizado tiene el mayor alcance limitado al tiempo del cronograma.

9 DISCUSIÓN

Se logra construir una red IoT para medir contaminación a pesar de contar con recursos económicos y de tiempo limitados. Con dos sensores fue posible verificar el correcto funcionamiento de la red. Sin embargo, existe la posibilidad que se presenten problemáticas o nuevos requerimientos si se incluye una mayor cantidad de nodos físicos. Se destacan los bajos costos para generar esta red de monitoreo, en comparación a la ya existente en el IBOCA. Permite que muchos ciudadanos sean partícipes de la generación de datos y evita que solo una institución centralizada como la secretaría de ambiente, cargue con toda la responsabilidad de la red de monitoreo.

El desempeño del trabajo realizado es satisfactorio para el límite de tiempo establecido. Sin embargo, se puede entrar a detallar de manera más profunda cada uno de los componentes de la red. Por el lado del nodo sensor es posible explorar otras tecnologías para sensar variables como contaminación por otro tipo de gases e inclusive contaminación acústica. La aplicación móvil tiene un gran potencial como herramienta de conciencia ambiental para los ciudadanos.

10 CONCLUSIONES

El proyecto de grado contribuye al desarrollo de soluciones IoT en la Universidad de los Andes ya que aún no existen énfasis en este tipo de soluciones. Es una gran integración de conocimientos de ingeniería electrónica e ingeniería de sistemas ya que se emplean conceptos desde la parte física de los sensores hasta el desarrollo de software y modelos de despliegue en la nube. Es por ello que se utilizó un gran número de herramientas como que en muchas ocasiones tienen una curva de aprendizaje que toma un tiempo considerable.

Es posible garantizar las especificaciones de un sistema IoT con las tecnologías y herramientas empleadas para el proyecto de grado. Si bien el prototipo cumple de manera satisfactoria con

algunas de ellas como lo es la tolerancia a fallas, existen otros atributos de calidad en los que se puede profundizar aún más. La ventaja es que esta profundización se puede hacer de manera fácil por medio de actualizaciones a la red ya existente. Por ejemplo, para incluir mayor seguridad es posible generar certificados y almacenarlos directamente en los nodos sensores por medio del sistema de archivos, por el lado del bróker es posible cambiar la configuración para exigir el uso de información cifrada en las comunicaciones.

Una solución IoT es viable para el monitoreo de la calidad del aire en Bogotá ya que puede iniciar con un pequeño número de nodos y puede escalar hasta el objetivo de cubrir la ciudad con aproximadamente 40 mil nodos. Organizaciones como universidades, conjuntos residenciales, zonas industriales o barrios pueden iniciar con la implementación de esta red de monitoreo para sus propias instalaciones. Posteriormente es posible incluir a ciudadanos de común que deseen obtener un nodo fijo para sus hogares o un nodo móvil para sus medios de transporte como bicicletas o automóviles. Finalmente es posible incluir a nivel de localidad o distrital diferentes organizaciones que desplieguen nodos por toda la ciudad, por ejemplo a la secretaría de ambiente o al SITP.

11 AGRADECIMIENTOS

Agradecimiento especial a mi asesor de proyecto de grado Fredy Enrique Segura y al co-asesor Juan José García por el apoyo en el desarrollo del proyecto de grado.

Agradezco a mis padres Ana Edith Siachoque y Wilson Hair Hernández por el apoyo en mi formación como persona.

A todos los profesores, estudiantes y demás miembros de la Universidad de los Andes que me apoyaron a lo largo de mi carrera universitaria.

12 REFERENCIAS

- [1] Milán F. M. Diseño e implementación de un sistema de medida de gases con Arduino. Escuela Universitaria Politécnica de Teruel. Tomado de: <https://zaguan.unizar.es/record/59102/files/TAZ-TFG-2016-2689.pdf>
- [2] S.S.Solapurueand H.Kenchannavar. Internet of Things: A survey related to various recent architectures and platforms available.
- [3] Muladi Muladi, Siti Sendari, Triyanna Widyaningtyas. Outdoor Air Quality Monitor Using MQTT Protocol on Smart Campus Network. Universitas Negeri Malang
- [4] Vrushali Ladekar, Dr.Rohin Daruwala. INDOOR AIR QUALITY MONITORING ON AWS USING MQTT PROTOCOL. 10th ICCCNT 2019. Universitas Negeri Malang
- [5] Pearl Pullan, Chitra Gautam, Vandana Niranjan. Air Quality Management System. 2020 IEEE International Conference on Computing, Power and Communication Technologies (GUCON). Galgotias University, Greater Noida, UP, India.
- [6] Sebastian Arturo Rodriguez, Juan Camilo Tangarife Palacio. Internet of things on smart cities. Universidad de los Andes

- [7] Espressif Systems (Shanghai) Co., Ltd. ESP-IDF Programming Guide. Tomado de <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/index.html>
- [8] Flutter documentation Tomado de <https://docs.flutter.dev>
- [9] HANWEI ELECTRONICS CO ., LTD. Techinical data MQ-7 GAS SENSOR. Tomado de: <https://www.sparkfun.com/datasheets/Sensors/Biometric/MQ-7.pdf>
- [10] HANWEI ELECTRONICS CO ., LTD. Techinical data MQ-135 GAS SENSOR. Tomado de https://www.electronicoscaldas.com/datasheet/MQ-135_Hanwei.pdf
- [11] Sharp. GP2Y1010AU0F datasheet. Tomado de: https://global.sharp/products/device/lineup/data/pdf/datasheet/gp2y1010au_e.pdf
- [12] MQTT.gorg. MQTT Publish / Subscribe Architecture Tomado se: <https://mqtt.org>
- [13] Gouri. Getting Started with ESP32 Bluetooth Low Energy (BLE) on Arduino IDE. Tomado de: <https://www.electrorules.com/getting-started-with-esp32-bluetooth-low-energy-ble-on-arduino-ide/>

13 APENDICES

Propuesta inicial del proyecto.

Especificaciones adicionales de notación o conceptos usados en el trabajo.

Facturas de materiales y proveedores con dirección y números de catalogo.

Instrumentos usados con información de identificación.

Documentación adicional de procesos intermedios en el diseño o validación del proyecto.